UPF EDITORA
UNIVERSIDADE DE PASSO FUNDO

RBCA
ISSN 2176-6649

Revista Brasileira de Computação Aplicada, Novembro, 2021

DOI: 10.5335/rbca.v13i3.12653
Vol. 13, Nº 3, pp. 86–100
Homepage: seer.upf.br/index.php/rbca/index

ARTIGO ORIGINAL

AutoRL-TSP-RSM: sistema de aprendizado por reforço automatizado com metodologia de superfície de resposta para o problema do caixeiro viajante

AutoRL-TSP-RSM: automated reinforcement learning system with response surface methodology for the traveling salesman problem

Gleice Kelly Barbosa Souza[1] and André Luiz Carvalho Ottoni [1]

[1]Centro de Ciências Exatas e Tecnológicas (CETEC), Universidade Federal do Recôncavo da Bahia (UFRB)

*kelly.189@hotmail.com; andre.ottoni@ufrb.edu.br

Recebido: 11/06/2021. **Revisado:** 17/11/2021. **Aceito:** 29/11/2021.

Resumo

A definição de parâmetros é uma importante etapa para a utilização de métodos de Aprendizado de Máquina. No entanto, pode ser altamente custoso definir esses valores de condições iniciais para cada aplicação. Assim, este trabalho tem como objetivo propor um sistema de Aprendizado de Máquina Automatizado para ajuste de parâmetros. Nesta linha, foi desenvolvido um método de Aprendizado por Reforço Automatizado aplicado ao Problema do Caixeiro Viajante. O sistema proposto ajustou através da Metodologia de Superfície de Resposta dois parâmetros (taxa de aprendizado e fator de desconto) do algoritmo Q-learning. Os resultados revelaram que os valores ajustados pelo método proposto alcançaram, em geral, as melhores soluções, em comparação com a adoção de parâmetros da literatura.

Palavras-Chave: Aprendizado por Reforço; AutoML; Problema do Caixeiro Viajante.

Abstract

The tuning of parameters is an important step towards the use of machine learning methods. However, it can be costly to define these initial condition values for each application. Thus, this paper aims to propose an Automated Machine Learning system for parameter tuning. In this line, an Automated Reinforcement Learning method was developed applied to the Traveling Salesman Problem. The proposed system adjusted through the Response Surface Methodology two parameters (learning rate and discount factor) of the Q-learning algorithm. The results revealed that the values adjusted by the proposed method reached, in general, the best solutions, in comparison with the adoption of parameters from the literature.

Keywords: AutoML; Reinforcement Learning; Traveling Salesman Problem.

1 Introdução

O Aprendizado de Máquina, em inglês, *Machine Learning* (ML), é uma área multidisciplinar que envolve temas como

Inteligência Artificial (IA), probabilidade, estatística, complexidade computacional e psicologia (Mitchell, 1997, Sutton and Barto, 2018). Em ML, o objetivo é o desenvolvimento de sistemas capazes de aprender com suas experiências, de forma a melhorar no desempenho da execução da tarefa que foi designada (Monard and Baranauskas, 2003, Celiberto Jr, 2007, Stange, 2011, Russell and Norvig, 2013). Nesse aspecto, técnicas de ML podem ser aplicadas em diversos contextos, como: classificação, análise de dados, robótica, jogos e reconhecimento de padrões (Mitchell, 1997, Bianchi, 2004, Serra, 2004, Rossi, 2015, Sutton and Barto, 2018, Boeing et al., 2019, Alzubaidi et al., 2021).

Uma dos grandes desafios para o bom desempenho na utilização de métodos de aprendizado de máquina é a definição de quais algoritmos e parâmetros serão utilizados durante os experimentos (Brazdil et al., 2008, Hutter et al., 2018, Tuggener et al., 2019). A complexidade desta etapa pode ser explicada pelo fato que para cada situação pode existir uma configuração específica de algoritmo e/ou parâmetro que irá fornecer melhores resultados (Feurer et al., 2015, Makmal et al., 2016, Mantovani et al., 2019, Cai et al., 2020). Assim, diversos métodos já foram propostos na literatura para o ajuste dessas configurações iniciais, dentre elas têm-se: Metodologia de Superfície de Resposta (RSM - *Response Surface Methodology*) (Ottoni, Nepomuceno and Oliveira, 2016, Ottoni et al., 2019, Lakshmi et al., 2020), Simulação Projetiva (Makmal et al., 2016), Métodos Empíricos (Gershman, 2016), Algoritmos Bioinspirados (Rossi, 2009), Algoritmo *Firefly* (Wang et al., 2017) e Algoritmos Genéticos (Espindola, 2009, Feitosa et al., 2009, Zhao et al., 2012).

Nesse sentido, de forma a automatizar o processo de seleção parâmetros e algoritmos de *ML* surgiu o *Automated Machine Learning* (AutoML). Os sistemas de AutoML, podem ser utilizados sob duas vertentes, a da recomendação (Brazdil et al., 2008, Mantovani et al., 2015, Cai et al., 2020) e a da otimização (Hutter et al., 2018, Tsiakmaki et al., 2019, Stamoulis et al., 2020). Uma das principais contribuições dos sistemas de AutoML encontra-se na redução do esforço empregado pelo experimentador para definir as condições adequadas para os métodos de ML, dado que, tais configurações são definidas automaticamente pelo sistema (Brazdil et al., 2008, Hutter et al., 2018).

Uma possível área de aplicação de AutoML é na definição de condições experimentais de Aprendizado por Reforço (RL) (Ottoni, Ottoni, Oliveira and Nepomuceno, 2020). No RL, o agente aprende a partir de interações com o ambiente, de forma a buscar as decisões que proporcionem os maiores retornos (Martins, 2007, Sutton and Barto, 2018, Bianchi, 2004, Serra, 2004, Santos, 2009, Santos et al., 2014, Sutton and Barto, 2018). Ao tomar essas decisões, o agente irá receber reforços (recompensa ou punição) que irão determinar a natureza da decisão tomada (Martins, 2007, Russell and Norvig, 2013, Santos et al., 2014). O RL pode ser aplicado nas mais diversas situações, dentre elas: jogos (Russell and Norvig, 2013, Sutton and Barto, 2018), robótica (Bianchi, 2004, Goldbarg and Luna, 2005, Martins, 2007, Russell and Norvig, 2013, Ottoni, 2016, Ottoni, Nepomuceno and Oliveira, 2016, Sutton and Barto, 2018), sistemas multiagentes (Ottoni, 2016, Ottoni, Nepomuceno and Oliveira, 2016). Outra importante área de aplicação de Aprendizado por Reforço é na otimização combinatória (Santos, 2009, Ottoni et al., 2017, Alipour et al., 2018, Ottoni, Nepomuceno, de Oliveira and de Oliveira, 2020).

Nesse aspecto, trabalhos recentes na literatura apresentaram metodologias para ajuste de parâmetros de Aprendizado por Reforço aplicado em um conhecido problema de otimização combinatória, o Problema do Caixeiro Viajante (PCV) (Ottoni et al., 2018, Ottoni, Ottoni, Oliveira and Nepomuceno, 2020). Ottoni et al. (2018) utilizam a Metodologia de Superfície de Resposta (RSM) para realizar a definição de dois parâmetros (taxa de aprendizado e fator de desconto) de RL aplicado no PCV. No entanto, em Ottoni et al. (2018) os parâmetros não foram gerados como um processo de AutoML. Por outro lado, Ottoni, Ottoni, Oliveira and Nepomuceno (2020) propõem um sistema de Aprendizado por Reforço automatizado (AutoRL) para o Problema do Caixeiro Viajante utilizando o método *Variable Neighborhood Search*. Dessa forma, o presente estudo busca avançar esses estudos recentes da literatura, automatizando o processo de geração de parâmetros com RSM.

Baseando-se na relevância da aplicação do Aprendizado por Reforço em problemas de otimização combinatória, como Problema do Caixeiro Viajante (Gambardella and Dorigo, 1995), Problema da Mochila (Ottoni et al., 2017) e *Sequential Ordering Problem* (Ottoni, Nepomuceno, de Oliveira and de Oliveira, 2020), o objetivo deste trabalho é propor um sistema de AutoRL aplicado ao Problema do Caixeiro Viajante. Para isso, será automatizado o processo de ajuste de parâmetros com Metodologia de Superfície de Resposta, utilizando a modelagem proposta por Ottoni et al. (2018).

Este artigo está divido em seções. A seção 2 demonstra conceitos teóricos. A seção 3, apresenta a metodologia empregada durante a construção deste trabalho. Já na seção 4, são apresentados os resultados. Por fim, na seção 5 é apresentada a conclusão.

2 Fundamentação Teórica

2.1 Aprendizado de Máquina

O Aprendizado de Máquina (ML) é um campo da Inteligência Artificial (IA) que pode ser dividido em três categorias (Russell and Norvig, 2013):

- **Aprendizado Supervisionado:** no aprendizado supervisionado, o agente recebe um conjunto de entradas e saídas que servem como um professor para lhe instruir sobre qual é a saída esperada para cada entrada (Russell and Norvig, 2013, Ottoni, 2016).
- **Aprendizado por Reforço:** no aprendizado por reforço, o agente não recebe informações prévias sobre o ambiente, ele deve aprender a medida que interage com o ambiente e os reforços recebidos devem definir quais serão suas próximas decisões (Martins, 2007, Russell and Norvig, 2013, Ottoni, 2016).
- **Aprendizado não Supervisionado:** no aprendizado não supervisionado, o agente recebe os dados de entrada, mas não recebe exemplos de como devem ser os dados de saída. O próprio agente deve realizar o agrupamento de dados que sejam semelhantes e que possivelmente pertençam a um mesmo grupo (Russell and Norvig, 2013).

2.1.1 Aprendizado por Reforço

Aprendizado por Reforço (RL) é uma das áreas que envolve IA e seu funcionamento se baseia em tentativas e erros (Santos et al., 2014). No RL, o agente realiza uma ação em um determinado estado e em seguida recebe um reforço indicando seu sucesso ou fracasso (Mitchell, 1997, Faria, 2000, Bianchi, 2004). Ao realizar estas ações, o agente interage com o ambiente e aprende sobre onde se encontra. Assim, o mesmo busca realizar ações que vão lhe render as melhores recompensas, independentemente destas serem a curto ou a longo prazo (Russell and Norvig, 2013, Ottoni, 2016, Sutton and Barto, 2018).

2.1.2 Processos de Decisão de Markov

Os Processos de Decisão de Markov (*Markov Decision Process* – MDP) são utilizados para tomadas de decisões sobre situações incertas (Pellegrini and Wainer, 2007). O RL, tem em sua base o MDP, dado que, no RL o agente não sabe qual resultado sua ação irá gerar, sabendo somente após a realização da ação (Sutton and Barto, 2018).

Os MDP, possuem um conjunto de variáveis para o controle do ambiente, dentre elas: ação, estado, recompensa e uma função probabilística para transição de estado (Pellegrini and Wainer, 2007). Estas variáveis podem ser representadas pela quádrupla (S, A, T, R) (Pellegrini and Wainer, 2007, Ottoni, 2016), sendo:

- S o conjunto de estados;
- A o conjunto de ações;
- T a função probabilística;
- R a recompensa.

2.1.3 Algoritmos de Aprendizado por Reforço

Diversos algoritmos podem ser aplicados ao RL, alguns deles são o método de Diferença Temporal, o Q-learning e o Sarsa. Sendo que, os dois últimos são baseados no método de Diferença Temporal (Ottoni, 2016, Sutton and Barto, 2018).

2.1.3.1 *Diferença Temporal.*

O método de Diferença Temporal utiliza a diferença de utilidades entre estados consecutivos (Russell and Norvig, 2013). Este método foi proposto por Sutton (1988) e sua função é apresentada a seguir (Russell and Norvig, 2013, Sutton and Barto, 2018):

$$U(s) = U(s) + \alpha[R(s) + \gamma U(s') - U(s)] \quad (1)$$

em que,

- $U(s)$ é a utilidade no estado atual;
- α é a taxa de aprendizado;
- γ é o fator de desconto;
- $R(s)$ é a recompensa;
- $U(s')$ é a utilidade no estado futuro.

2.1.3.2 *Q-learning.*

O Q-learning, é baseado no método de Diferença Temporal para situações em que não há modelo (Watkins and Dayan, 1992, Sutton and Barto, 2018). O Q-learning foi proposto por Watkins (1989) e sua função é exibida a seguir (Watkins and Dayan, 1992, Sutton and Barto, 2018):

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha[r(s_t, a_t) + \gamma max_{a'} Q(s', a') - Q_t(s_t, a_t)] \quad (2)$$

em que,

- Q é a matriz de aprendizado;
- $r(s,a)$ é o reforço recebido por realizar a ação a_t no estado s_t;
- α é a taxa de aprendizado;
- γ é o fator de desconto;
- s_t é o estado atual;
- a_t é a ação realizada no estado atual;
- s' é o estado futuro;
- a' é a ação que será realizada no estado futuro.

2.1.3.3 *Sarsa.*

O Sarsa é uma modificação do Q-learning e seu nome foi dado por Sutton (1996). A modificação realizada no Q-learning se deu na forma em que a recompensa futura é concebida, enquanto o Q-learning recebe a recompensa máxima o Sarsa recebe a recompensa esperada (Sutton and Barto, 2018). Assim, sua equação é:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha[r(s_t, a_t) + \gamma Q(s', a') - Q_t(s_t, a_t)] \quad (3)$$

em que,

- Q é a matriz de aprendizado;
- $r(s,a)$ é o reforço recebido por realizar a ação a_t no estado s_t;
- α é a taxa de aprendizado;
- γ é o fator de desconto;
- s_t é o estado atual;
- a_t é a ação realizada no estado atual;
- s' é o estado futuro;
- a' é a ação que será realizada no estado futuro.

2.1.4 Parâmetros

- **Taxa de aprendizado**: a taxa de aprendizado (α) controla a velocidade com que o agente aprende. Este parâmetro pode variar entre 0 e 1. Sendo que, para os casos em que $\alpha = 0$ não existe aprendizado (Ottoni, Nepomuceno and Oliveira, 2016).
- **Fator de desconto**: o fator de desconto (γ) determina o quão importante é uma recompensa recebida pelo agente. Este parâmetro pode variar entre 0 e 1. Sendo que, para valores de γ próximos de 0, as recompensas são insignificantes. Em contrapartida, quanto mais próximo de 1 for o valor de γ, mais significativa será a recompensa (Celiberto Jr, 2007, Martins, 2007, Russell and Norvig, 2013, Ottoni, 2016).
- **Política ϵ-greedy**: a política ϵ-greedy (ϵ) determina o quão aleatórias são as decisões tomadas pelo agente. Assim como os parâmetros apresentados anteriormente, o valor da política ϵ-greedy também pode variar entre 0 e 1 (Celiberto Jr, 2007, Martins, 2007, Ottoni, 2016). Como regra de atualização da política ϵ-greedy tem-se (Celiberto Jr, 2007, Ottoni, 2016):

$$\pi(s) = \begin{cases} a^*, \text{com probabilidade } 1 - \epsilon \\ a_a, \text{com probabilidade } \epsilon \end{cases}$$

em que,

- $\pi(s)$ é a política para o estado atual;
- a^* é a melhor ação disponível para o estado atual;
- a_a é uma ação aleatória entre as disponíveis para o estado atual.

2.2 Problema do Caixeiro Viajante

2.2.1 *Explicação do Problema*

O Problema do Caixeiro Viajante (PCV) é constituído de um conjunto de N cidades que o agente deve visitar respeitando a restrição de passar somente uma vez por cada cidade. Com exceção da cidade final, a qual deve ser a mesma da inicial (Bodin, 1983, Pedro, 2013, Santos, 2014, Silva, 2014, Vitor, 2015, Ottoni et al., 2015, Ottoni, Nepomuceno and de Oliveira, 2016, Ottoni, Nepomuceno and Oliveira, 2016). Além disso, este problema pode receber a classificação de simétrico ou assimétrico. Para o caso simétrico, o custo final da trajetória realizada pelo agente não depende do sentido adotado. Em contrapartida, no caso assimétrico, o custo final está associado ao sentido adotado para à realização do percurso (Pedro, 2013, Santos, 2014, Ottoni, Nepomuceno and de Oliveira, 2016, Ottoni, Nepomuceno and Oliveira, 2016).

Algumas das aplicações deste problema são apresentadas por Goldbarg and Luna (2005) e Vitor (2015), dentre elas têm-se:

- Problemas de roteamento de veículos.
- Otimização do movimento de ferramentas de corte.
- Perfuração de placas de circuito impresso.
- Solução de problemas de sequenciamento de tarefas.
- Cortes em chapas de aço e vidro.

2.2.2 *Formulação*

A formulação descrita abaixo foi apresentada em Bodin (1983). No entanto, existem diversas formulações para o PCV e algumas delas podem ser encontradas em Goldbarg and Luna (2005) e Bodin (1983).

$$Minimizar \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} x_{ij} \tag{4}$$

sujeito a:

$$\sum_{i=1}^{N} x_{ij} = 1 \qquad (\forall j = 1, \ldots, N) \tag{5}$$

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad (\forall i = 1, \ldots, N) \tag{6}$$

$$x_{ij} \in \{0, 1\} \qquad (\forall i, j = 1, \ldots, N) \tag{7}$$

$$X = x_{ij} \in S \qquad (\forall i, j = 1, \ldots, N) \tag{8}$$

A Eq. (4) representa a função objetivo do PCV. Nesta, c_{ij} representa a distância entre os vértices i e j. x_{ij} indica se a aresta entre os vértices i e j faz parte da solução do problema. Em caso positivo, x_{ij} assume o valor 1. No entanto, caso a aresta não faça parte da solução, x_{ij} assume o valor 0. Além disto, as equações Eq. (5) e Eq. (6) garantem que cada vértice seja visitado pelo agente somente uma vez. A Eq. (7) garante que a variável x_{ij} só possa assumir valores binários. Por fim, a Eq. (8) garante que não serão formadas sub-rotas para solução do problema (Bodin, 1983, Vitor, 2015, Ottoni, 2016, Ottoni, Ottoni, Oliveira and Nepomuceno, 2020).

2.3 AutoML

Dois dos principais pontos a serem definidos durante o Aprendizado de Máquina é qual algoritmo será utilizado e quais parâmetros este algoritmo irá receber (Tuggener et al., 2019). Normalmente, estas configurações iniciais são realizadas pelo usuário, porém para cada situação pode haver uma configuração específica que irá gerar um resultado melhor (Feurer et al., 2015). Assim, diversas configurações são testadas com o intuito de encontrar as melhores possíveis. No entanto, a definição dessas configurações inciais pode não ser tão óbvia e o experimentador pode não encontrar as configurações que forneceriam os melhores resultados (Mantovani et al., 2016). Neste sentido, tem-se o AutoML (*Automated Machine Learning*), que possui como finalidade principal a definição automatizada dos parâmetros e/ou algoritmos. Alguns de seus propósitos são (Brazdil et al., 2008, Hutter et al., 2018):

- Diminuição do esforço empregado pelo usuário durante a Aprendizagem de Máquina.
- Auxiliar o usuário durante a escolha do algoritmo adequado para cada problema.
- Aperfeiçoamento de desempenho dos algoritmos de Aprendizado de Máquina, dado que os parâmetros são ajustados de acordo com o problema.

O AutoML possui duas abordagens, a da otimização (Chen et al., 2019, Stamoulis et al., 2020) e a da recomendação (Mantovani et al., 2019, Cai et al., 2020). Um sistema AutoML que faz uso da recomendação é capaz de auxiliar o usuário durante a escolha de qual algoritmo deve ser utilizado (Brazdil et al., 2008, Mantovani et al., 2019). A depender do sistema desenvolvido, ele pode ainda sugerir ao usuário quais parâmetros devem ser aplicados, dado que os parâmetros podem interferir no desempenho do algoritmo (Brazdil et al., 2008, Hutter et al., 2014, Feurer et al., 2015, Mantovani et al., 2015, 2016, 2019). Já no caso de um sistema AutoML que utiliza a otimização, este sistema irá ajustar os parâmetros a fim de obter os que irão proporcionar o melhor resultado possível (Brazdil et al., 2008, Mantovani et al., 2015, Hutter et al., 2018).

Por conta de alguns fatores aplicar o AutoML pode ser complexo na prática (Brazdil et al., 2008, Mantovani et al.,

2016, Hutter et al., 2018, Mantovani et al., 2019):

- A avaliação de grandes conjuntos de dados pode ser complexa.
- O desenvolvimento de um sistema AutoML pode tornar-se complexo.
- As configurações podem levar muito tempo para serem avaliadas.

Um dos principais assuntos dentro do Aprendizado de Máquina automatizado é o meta-aprendizado (*metalearning*) (Brazdil et al., 2008, Feurer et al., 2015, Hutter et al., 2018, Tuggener et al., 2019, Mantovani et al., 2019). No *metalearning* o sistema utiliza sua experiência adquirida em tarefas já realizadas para realizar novas tarefas (Feurer et al., 2015). Isto faz com que o esforço necessário para realizar a tarefa atual seja inferior ao esforço empregado para realizar as tarefas anteriores (Brazdil et al., 2008, Hutter et al., 2018). Não menos importante, um ponto a se ter atenção é que quanto maior a similaridade entre as tarefas (já realizadas e a serem realizadas), maiores são as chances de obter bons resultados (Hutter et al., 2018, Tuggener et al., 2019).

3 Metodologia

Nesta seção será apresentada a metodologia utilizada durante o desenvolvimento deste trabalho. A primeira subseção apresenta a modelagem de Aprendizado por Reforço selecionada para o Problema do Caixeiro Viajante. Já na subseção seguinte, é abordado acerca da Metodologia de Superfície de Resposta, método utilizado para implementar o AutoML. Na sequência, são apresentadas as etapas realizadas durante o experimento. Por fim, a última subseção apresenta o algoritmo de AutoML proposto por este trabalho.

3.1 Modelagem do AR aplicado ao PCV

Para solucionar o Problema do Caixeiro Viajante (PCV) através do Aprendizado por Reforço é necessário definir quais serão os estados, as ações e os reforços. Assim, após a análise de alguns trabalhos da literatura estes pontos foram definidos da seguinte maneira (Gambardella and Dorigo, 1995, Bianchi et al., 2009, Santos, 2009, Júnior et al., 2010, Ottoni, Ottoni, Oliveira and Nepomuceno, 2020):

- **Estados:** são todos os nós (cidades) do problema selecionado.
- **Ações:** são os estados ainda não visitados no estado atual.
- **Reforços:** após executar uma ação o agente irá receber um reforço que seguirá a seguinte regra:

$$r_{ij} = -d_{ij}, \tag{9}$$

sendo que, r_{ij} representa o reforço recebido pela transição do estado i para o estado j e d_{ij} representa a distância entre os respectivos estados (Ottoni, 2016).

3.2 Metodologia de Superfície de Resposta

A Metodologia de Superfície de Resposta, do inglês *Response Surface Methodology* (RSM), é uma técnica matemática que permite à aproximação de funções. Através do RSM, é possível otimizar processos que possuem múltiplos fatores que influenciam no resultado final. Geralmente, utiliza-se modelos de primeira ou segunda ordem para aproximar estas funções (Myers et al., 2016), conforme Eq. (10) e Eq. (11):

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon \tag{10}$$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 + \varepsilon \tag{11}$$

A Eq. (10) representa o modelo de primeira ordem. Já Eq. (11) representa o modelo de segunda ordem. Nestas equações, β_n são os coeficientes, x_1 e x_2 são as variáveis independentes, y é a variável resposta e ε é o erro associado ao modelo ajustado (Ottoni et al., 2019).

Como foi proposto em Ottoni et al. (2018), neste trabalho será adotado o modelo de segunda ordem. Dado que, este modelo foi adotado em outros trabalhos na literatura e apresentou bons resultados. Após o ajuste da Eq. (11), a Eq. (12) representa a equação que será utilizada:

$$y = \beta_0 + \beta_1 \alpha + \beta_2 \gamma + \beta_3 \alpha^2 + \beta_4 \gamma^2 + \beta_5 \alpha\gamma + \varepsilon \tag{12}$$

na qual,

- $x_1 = \alpha$;
- $x_2 = \gamma$;
- y é a variável resposta que irá conter a distância total do trajeto realizado pelo agente.

3.3 Planejamento dos Experimentos

Inicialmente, foi definida qual modelagem do Aprendizado por Reforço seria aplicada ao Problema do Caixeiro Viajante juntamente com a linguagem que seria utilizada para desenvolver o sistema e qual algoritmo seria utilizado no mesmo. Assim, foi selecionado o algoritmo de Q-learning e este foi desenvolvido utilizando a linguagem R. Em seguida, foi selecionado qual valor de política ϵ-greedy (ϵ) seria utilizado e quais valores de Taxa de Aprendizado (α) e Fator de Desconto (γ) seriam utilizados para realização das combinações entre os parâmetros que seriam aplicados durante a primeira etapa do experimento. Os parâmetros (Taxa de Aprendizado e Fator de Desconto) selecionados para esta etapa inicial, foram aplicados em Ottoni (2016) e Ottoni et al. (2018). A Tabela 1 reúne estes parâmetros. Além disso, a mesma apresenta ainda algumas informações adicionais sobre a primeira etapa.

Já à Tabela 2, contém os dados pré-definidos da etapa final. Visando ajustar as configurações iniciais desta última etapa, a quantidade de episódios, de valores de parâmetros e combinações realizadas foram alterados. Entretanto, as

Tabela 1: Dados iniciais da primeira etapa do experimento.

	Quantidade	Valores
α	8	0,01; 0,15; 0,30; 0,45; 0,60; 0,75; 0,90; 0,99
γ	8	0,01; 0,15; 0,30; 0,45; 0,60; 0,75; 0,90; 0,99
ϵ	1	0,01
Combinações	$8 \times 8 \times 1 = 64$	-
Épocas por Combinação	5	-
Episódios por Época	1000	-
Episódios por Combinação	$5 \times 1000 = 5000$	-
Total de Épocas	$5 \times 64 = 320$	-
Total de Episódios	$1000 \times 320 = 320000$	-

demais configurações não necessitaram de ajuste, sendo assim, as mesmas foram mantidas.

Tabela 2: Dados iniciais da etapa final do experimento.

	Quantidade	Valores
α	1	-
γ	1	-
ϵ	1	0,01
Combinações	$1 \times 1 \times 1 = 1$	-
Épocas por Combinação	5	-
Episódios por Época	10000	-
Episódios por Combinação	$5 \times 10000 = 50000$	-
Total de Épocas	5	-
Total de Episódios	$5 \times 10000 = 50000$	-

As instâncias selecionadas para o experimento foram obtidas na biblioteca TSPLIB[1] (Reinelt, 1991). Na TSPLIB podem ser encontrados diversos problemas de otimização, a biblioteca oferece dados do Problema do Caixeiro Viajante e de outros problemas relacionados a ele (Reinelt, 1995, Bianchi, 2004, Santos, 2014). Alguns dos problemas que podem ser encontrados na TSPLIB são seguintes seguimentos (Reinelt, 1995):

- Symmetric traveling salesman problem (TSP): problemas do caixeiro viajante simétrico.
- Asymmetric traveling salesman problem (ATSP): problemas do caixeiro viajante assimétrico.

Assim, foram selecionados problemas simétricos e assimétricos do PCV. Tais problemas estão reunidos na Tabela 3.

Tabela 3: Problemas selecionados da TSPLIB.

Tipo	Problema	Cidades	Ótimo
TSP	swiss42	42	1273
	eil51	51	426
	berlin52	52	7542
	st70	70	675
	eil76	76	538
	pr76	76	108159
	rat99	99	1211
	kroa100	100	21282
	eil101	101	629
	bier127	127	118282
	ch130	130	6110
ATSP	ftv33	34	1286
	p43	43	5620
	ftv44	45	1613
	ftv47	48	1776
	ry48p	48	14422
	ft53	53	6905
	ftv64	65	1839
	ft70	70	38673

3.4 Algoritmo Proposto

O sistema de AutoML proposto, denominado AutoRL-TSP-RSM, foi implementado na linguagem R. Neste, foi realizada a implementação manual de grande parte dos processos realizados, como do algoritmo Q-learning por exemplo. Associado a isto, fez-se ainda o uso de bibliotecas e funções disponíveis na linguagem R, como por exemplo a biblioteca '*rsm*' e as funções *anova, summary* e *ks.test*. Não menos importante, vale ressaltar que o algoritmo desenvolvido faz uso de todas as informações já apresentadas anteriormente, nas etapas inicial e final é utilizada a modelagem do Aprendizado por Reforço selecionada para o problema. Já para à automatização, é aplicada a Metodologia de Superfície de Resposta. O sistema desenvolvido pode ser utilizado tanto com instâncias simétricas quanto com instâncias assimétricas. O método AutoRL-TSP-RSM utiliza como métrica durante a modelagem o menor valor de distância de rota por época alcançado dentre os resultados da primeira etapa.

O Algoritmo 1 descreve o algoritmo Q-learning que foi implementado, este algoritmo é utilizado durante a primeira e a última etapa. Na primeira etapa, os valores de fator de desconto e taxa de aprendizado recebidos pela função são apresentados na Tabela 1. Já para à última etapa, os parâmetros serão definidos pelo sistema utilizando a função descrita no Algoritmo 2.

O método utilizado para à automatização do sistema é a Metodologia de Superfície de Resposta. Inicialmente, é gerado o modelo de regressão linear múltipla, em seguida alguns dados são extraídos do modelo ajustado para à realização de algumas validações. Em uma destas verificações avalia-se os resíduos do modelo gerado, assim, esta validação tem como objetivo verificar se os resíduos do modelo seguem uma distribuição normal. Além disto, é realizada também a verificação do nível de significância do modelo ajustado. Não menos importante, avalia-se ainda se os valores obtidos para α e γ utilizando RSM respeitam os

[1] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

Algoritmo 1: Função implementada do Q-learning

```
1  Função Q-Learning(α, γ, ϵ, numeroDeEpisodios,
     tamanhoDaInstancia):
2      contadorEpisodios: 1
3      Em cada s,a faça Q(s,a) = 0
4      enquanto (contadorEpisodios < numeroDeEpisodios)
         faça
5          contadorEstados: 1
6          Observe o estado s
7          enquanto (contadorEstados <
             tamanhoDaInstancia) faça
8              Selecione a ação a utilizando a política
                 ϵ-greedy
9              Execute a ação a
10             Receba a recompensa imediata r(s,a)
11             Observe o novo estado s'
12             Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + α[r(s_t, a_t) +
                 γmax_{a'} Q(s', a') − Q_t(s_t, a_t)]
13             s = s'
14         fim
15     fim
```

Algoritmo 2: Modelagem com RSM e definição dos pontos estacionários

```
1  Função RSM():
2      Ajuste dos dados para um modelo de regressão
3      Geração de dados estatísticos do modelo
4      Geração da Superfície de Resposta utilizando o RSM
5      Definição dos pontos estacionários do Modelo RSM
6      α e γ: Pontos estacionários do modelo
7      se (Normalidade > 0,05 e Significância < 0,05 e α >
         0,01 e α < 1 e γ > 0 e γ < 1) então
8          Sistema de otimização
9      senão
10         Sistema de recomendação
11     fim
12     Defina os novos parâmetros: α e γ
```

limites definidos (α e γ devem variar somente entre 0 e 1). Caso todas verificações sejam satisfeitas, o sistema seguirá a vertente da otimização e utilizará na etapa final os pontos estacionários da superfície gerada como parâmetros. Do contrário, o sistema seguirá a vertente da recomendação e utilizará na etapa final os parâmetros que proporcionaram o melhor desempenho durante a etapa inicial. O Algoritmo 2 apresenta estas etapas.

Finalmente, o Algoritmo 3 faz uso dos Algoritmos 1 e 2, sendo assim, o mesmo representa o sistema implementado. Primeiramente, são definidos os dados que serão fixos durante a execução do sistema. A primeira etapa utiliza o Algoritmo 1 e os dados obtidos nesta etapa são armazenados para serem utilizados posteriormente pelo sistema nas etapas seguintes. Já na segunda etapa, é utilizado o Algoritmo 2 e através dele são definidos os novos valores de taxa de aprendizado e fator de desconto que serão aplicados na etapa final. Por último, o sistema utiliza novamente o algoritmo 1, no entanto, nesta etapa os valores de fator de desconto e taxa de aprendizado serão os

Algoritmo 3: Algoritmo AutoRL-TSP-RSM.

```
1  Defina a instância que será executada
2  Defina o parâmetro: ϵ
3  tamanhoDaInstancia: Número de cidades do problema
     selecionado
4  numeroEpocas: 5
                    Etapa 1
5  Defina os parâmetros: α e γ
6  numeroDeEpisodios: 1000
7  para cada α_t em α faça
8      para cada γ_t em γ faça
9          para cada epoca em numeroEpocas faça
10             Q-Learning(α, γ, ϵ, numeroDeEpisodios,
                 tamanhoDaInstancia)
11         fim
12     fim
13 fim
                    Etapa 2
14 RSM()
                    Etapa 3
15 numeroDeEpisodios: 10000
16 para cada epoca em numeroEpocas faça
17     Q-Learning(α, γ, ϵ, numeroDeEpisodios,
         tamanhoDaInstancia)
18 fim
```

definidos automaticamente na etapa anterior pelo próprio sistema.

3.5 Análise dos Resultados

A seção seguinte apresentará os resultados obtidos com este trabalho. Para tal, segue-se as seguintes etapas:

i. Análise das Medidas de Adequação obtidas. Neste trabalho irá analisar-se os coeficientes de determinação múltipla, o nível de significância e o índice de normalidade.
ii. Verificação dos Coeficientes Ajustados. Será verificado o nível de significância obtido para cada coeficiente ajustado para à Eq. (12).
iii. Análise dos Parâmetros Ajustados. Nesta seção serão apresentados os parâmetros ajustados pelo sistema proposto. Além disto, é avaliado se existe alguma relação que parametrize a definição dos parâmetros.
iv. Comparação com parâmetros da literatura. Nesta seção serão apresentados os resultados obtidos ao aplicar os parâmetros ajustados pelo sistema e ao aplicar alguns parâmetros da literatura. Além disso, será realizada uma comparação entre estes resultados.

4 Resultados

Esta seção apresenta os resultados obtidos neste trabalho. Inicialmente, são apresentadas as medidas de adequação obtidas durante o experimento. Na sequência, são apre-

sentados os coeficientes ajustados durante o experimento. Em seguida, apresenta-se os parâmetros ajustados pelo sistema proposto. Por fim, é apresentada uma comparação realizada entre resultados obtidos ao aplicar os parâmetros ajustados pelo sistema proposto e os parâmetros obtidos da literatura.

4.1 Medidas de Adequação

Como forma de análise dos modelos gerados pelo sistema, algumas propriedades de cada instância testada foram avaliadas. Assim, foram extraídos de cada modelo a normalidade, a significância, o coeficiente de determinação múltipla (R_a) e o coeficiente de determinação múltipla ajustado (R_a^2). Todos estes dados foram extraídos utilizando a função *sumamry* do R.

Os coeficientes R_a e R_a^2, expressam o quão bom é o modelo ajustado e o valor de ambas as métricas deve variar entre 0 e 1. Assim, quanto maior for o valor destas métricas, melhor é o modelo. Contudo, o valor de R_a pode aumentar por conta da adição de novos termos no modelo. Desta forma, faz-se necessário também a avaliação de R_a^2 para verificar a qualidade do modelo ajustado (Myers et al., 2016).

Para avaliar a normalidade dos resíduos dos modelos, foi selecionado o teste de Kolmogorov-Smirnov (KS). Uma vez que o mesmo é dos testes mais utilizados para verificar se os dados seguem uma distribuição normal (Leotti et al., 2005, Razali et al., 2011). Avaliar a normalidade dos dados é de extrema importância, sendo que, quando não é possível comprovar a normalidade dos dados, pode ser que estes dados não sejam confiáveis. O teste KS avalia hipóteses e define se a hipótese é verdadeira ou não, assim, têm-se a hipótese inicial (H_0) e a hipótese alternativa (H_a) (Razali et al., 2011):

$$\begin{cases} H_0 : \text{Os dados seguem uma distribuição normal.} \\ H_a : \text{Os dados não seguem uma distribuição normal.} \end{cases}$$

O índice de normalidade dos resíduos dos modelos foi definido como maior que 5%. Neste caso, aceita-se H_0 e os dados possuem distribuição normal. Do contrário, aceita-se H_a ($p - valor_{KS} < 0{,}05$) e os dados não possuem distribuição normal.

Assim como a avaliação da normalidade, a avaliação da significância também é realizada por meio de hipóteses. Igualmente, têm-se a hipótese inicial (H_0) e a hipótese alternativa (H_a). O índice de significância indica se as variáveis possuem relação com a variável resposta do modelo ajustado (Myers et al., 2016).

$$\begin{cases} H_0 : \text{O não modelo é significativo.} \\ H_a : \text{O modelo é significativo.} \end{cases}$$

O nível de significância foi definido como 5%, neste caso o modelo é significativo e aceita-se H_a. Já em caso contrário, o modelo é dito não significativo e aceita-se H_0 ($p\text{-}valor > 0{,}05$).

A Tabela 4 apresenta as medidas de adequação obtidas para cada instância utilizada. Assim, na mesma pode-se verificar que as instâncias swiss42, eil51, berlin52, st70, eil76, rat99, kroa100, eil101, bier127, ftv33, ftv44, ftv47, ry48p, ft53, ftv64 e ftv70 atenderam as condições que foram avaliadas durante o experimento. Dessa forma, durante a etapa final foram aplicados os pontos estacionários da superfície de resposta como parâmetros. Em contrapartida, as instâncias pr76, ch130 e p43 não atenderam aos critérios definidos, consequentemente, o sistema aplicou na etapa final os parâmetros que proporcionaram os melhores resultados de distância de rota durante a primeira etapa.

Tabela 4: Medidas de adequação obtidas durante o experimento.

Tipo	Problema	$p - valor_{KS}$	p-valor	R^2	R_a^2
TSP	swiss42	0,149	0,000	0,728	0,723
	eil51	0,430	0,000	0,676	0,671
	berlin52	0,063	0,000	0,708	0,703
	st70	0,065	0,000	0,709	0,704
	eil76	0,404	0,000	0,726	0,722
	pr76	0,000	0,000	0,621	0,615
	rat99	0,707	0,000	0,689	0,684
	kroa100	0,058	0,000	0,762	0,758
	eil101	0,683	0,000	0,761	0,758
	bier127	0,526	0,000	0,792	0,788
	ch130	0,035	0,000	0,796	0,793
ATSP	ftv33	0,322	0,000	0,704	0,699
	p43	0,026	0,000	0,496	0,488
	ftv44	0,369	0,000	0,651	0,645
	ftv47	0,547	0,000	0,686	0,681
	ry48p	0,065	0,000	0,690	0,685
	ft53	0,534	0,000	0,707	0,400
	ftv64	0,794	0,000	0,700	0,695
	ft70	0,290	0,000	0,788	0,785

4.2 Coeficientes Ajustados

A Tabela 5 reúne os coeficientes ajustados durante o experimento para à Eq. (12) de todos os problemas utilizados durante o experimento. Nesta, os coeficientes são representados por β_0, β_1, β_2, β_3, β_4 e β_5. Não menos importante, ressalta-se que o nível de significância desejado é o mesmo definido na seção anterior.

A Tabela 6 por sua vez, apresenta as significâncias obtidas para cada coeficiente ajustado. Sendo que, p_0 indica a significância de β_0, p_1 a significância de β_1 e assim continuamente até que finalmente p_5 indica a significância de β_5. Assim como na seção anterior, os dados desta seção também foram obtidos utilizando a função *sumamry* do R.

À aplicação da Metodologia de Superfície de Resposta permite realizar análises através de gráficos dos dados de estudo. Estas análises podem ser realizadas em gráficos de contorno ou em superfícies de resposta. Sendo que, no caso dos gráficos de contorno utiliza-se o espaço bidimensional e nas superfícies de respostas utiliza-se o espaço tridimensional (Myers et al., 2016).

Diante do dados apresentados na Tabela 6, percebe-se que os coeficientes β_0, β_1 e β_3 cumpriram o nível de significância desejado, logo, pode-se constatar que estes coeficientes são significantes para os modelos ajustados. Já os demais coeficientes, obtiveram $p\text{-}valor > 0{,}05$ em alguns dos modelos ajustados, no entanto, os mesmos foram mantidos com a finalidade de preservar um modelo padrão de sistema AutoML para o experimento.

Tabela 5: Coeficientes ajustados durante o experimento.

Problema	β_0	β_1	β_2	β_3	β_4	β_5
swiss42	2071,92	-2170,38	-249,51	1646,53	547,45	-127,11
eil51	710,96	-692,58	-152,25	520,01	295,15	-80,49
berlin52	12326,50	-13782,60	-495,60	10077,20	1397,70	712,60
st70	1212,51	-1675,09	-166,73	1290,83	422,63	-143,95
eil76	1004,72	-1390,44	-333,26	1059,62	577,51	-102,06
pr76	153855,00	-66994,00	-32407,00	45932,00	44100,00	6611,00
rat99	2098,29	-2111,59	-1047,57	1489,62	1410,95	186,59
kroa100	47983,00	-78735,00	-6775,00	60500,00	18084,00	-8185,00
eil101	1333,94	-1883,11	-332,40	1401,07	617,40	-156,20
bier127	251936,00	-397990,00	-30905,00	293326,00	74918,00	-4089,00
ch130	14443,70	-26097,20	-1101,90	19883,00	3610,70	-1468,40
ftv33	2045,37	-1998,34	33,23	1532,51	248,00	-79,03
p43	5676,19	-63,41	5,72	36,39	-2,46	15,31
ftv44	2976,40	-3376,92	-1007,73	2488,68	1511,80	-28,98
ftv47	3248,39	-3472,71	-911,58	2570,78	1575,64	-93,70
ry48p	22921,40	-23333,70	-4922,00	17365,10	9276,30	-1312,60
ft53	10792,90	-6736,60	-3538,20	4976,30	5090,10	-762,50
ftv64	3948,60	-5436,80	-1437,30	4098,20	2378,80	-359,10
ft70	47105,04	-13744,19	-3969,24	8616,60	9086,44	-91,43

Tabela 6: Significância dos coeficientes ajustados.

Problema	p_0	p_1	p_2	p_3	p_4	p_5
swiss42	0,0000	0,0000	0,0259	0,0000	0,0000	0,1175
eil51	0,0000	0,0000	0,0011	0,0000	0,0000	0,0172
berlin52	0,0000	0,0000	0,4580	0,0000	0,0180	0,1420
st70	0,0000	0,0000	0,0688	0,0000	0,0000	0,0308
eil76	0,0000	0,0000	0,0000	0,0000	0,0000	0,0688
pr76	0,0000	0,0000	0,0000	0,0000	0,0000	0,0884
rat99	0,0000	0,0000	0,0000	0,0000	0,0000	0,0700
kroa100	0,0000	0,0000	0,0731	0,0000	0,0000	0,0030
eil101	0,0000	0,0000	0,0005	0,0000	0,0000	0,0235
bier127	0,0000	0,0000	0,0648	0,0000	0,0000	0,7360
ch130	0,0000	0,0000	0,2955	0,0000	0,0001	0,0555
ftv33	0,0000	0,0000	0,7516	0,0000	0,0077	0,3005
p43	0,0000	0,0000	0,3222	0,0000	0,6284	0,0003
ftv44	0,0000	0,0000	0,0000	0,0000	0,0000	0,8570
ftv47	0,0000	0,0000	0,0000	0,0000	0,0000	0,5580
ry48p	0,0000	0,0000	0,0007	0,0000	0,0000	0,2119
ft53	0,0000	0,0000	0,0000	0,0000	0,0000	0,0544
ftv64	0,0000	0,0000	0,0000	0,0000	0,0000	0,1290
ft70	0,0000	0,0000	0,0000	0,0000	0,0000	0,8970

4.2.1 *Superfície de Resposta e Gráfico de contorno*

A Fig. 1 apresenta o gráfico de contorno do problema berlin52. Já a Fig. 2, mostra a superfície de resposta deste mesmo problema. Sendo que, ambos os gráficos foram obtidos durante o experimento com o algoritmo de AutoML proposto.

Na Fig. 1 e na Fig. 2, pode-se verificar para quais faixas de valores de taxa de aprendizado e fator de desconto a distância final da rota tende a ser minimizada. Sendo que, em ambos os gráficos estas regiões são representadas com tons mais avermelhados, neste caso, a distância final da rota tende a ser minimizada quando são aplicados aproximadamente os valores de $0,6 \leq \alpha \leq 0,8$ e $0 \leq \gamma \leq 0,3$.

4.3 Parâmetros Ajustados

Através da Metodologia de Superfície de Resposta é possível modelar diversos tipos de problemas e pode-se ainda obter os pontos estacionários de cada modelo gerado. Os pontos estacionários, são responsáveis por representar o local de mínimo, máximo ou ponto de sela do modelo (Riboldi and Nascimento, 1994). A Tabela 7, reúne os parâmetros ajustados para cada modelo gerado por meio da Metodologia de Superfície de Resposta. Para os casos em que o modelo atendeu medidas de adequação são apresentados os pontos estacionários das superfícies obtidas. Já nos casos em que o modelo gerado não atendeu à alguma das verificações, são apresentados os valores de Taxa de Aprendizado e Fator de desconto quê quando combinados proporcionaram o melhor resultado de distância total de rota na primeira etapa do experimento. Vale ressaltar quê estes valores foram determinados através do algoritmo de AutoRL proposto durante a segunda etapa do experimento.

Ao verificar a Tabela 7, pode-se perceber que os valores de Taxa de Aprendizado e Fator de desconto obtidos para cada problema analisado foi diferente. Sendo assim, pode-se perceber que não há um padrão para definição dos

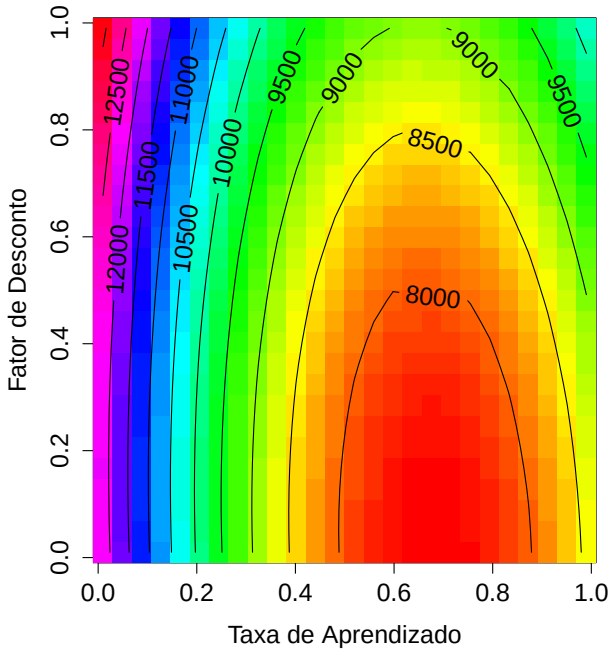

Figura 1: Gráfico de contorno do problema berlin52 gerado pelo algoritmo de AutoML proposto utilizando os parâmetros $\alpha = 0,684$ e $\gamma = 0,003$.

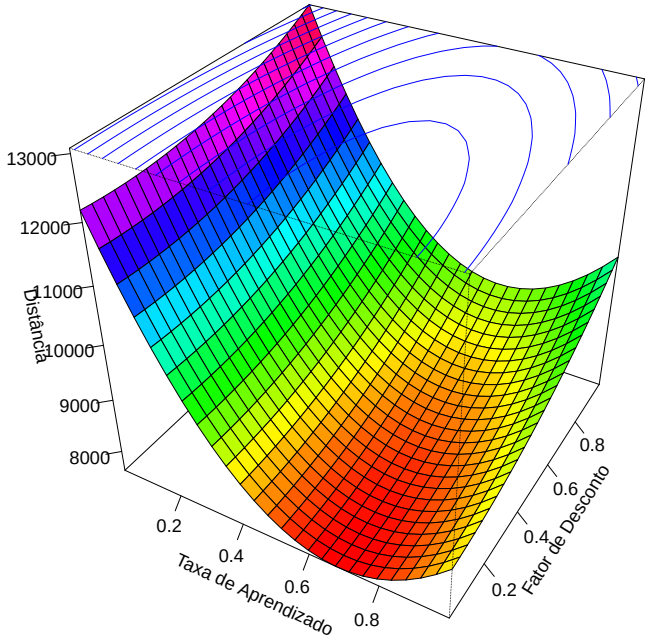

Figura 2: Superfície de resposta do problema berlin52 gerada pelo algoritmo de AutoML proposto utilizando os parâmetros $\alpha = 0,684$ e $\gamma = 0,003$.

Tabela 7: Parâmetros ajustados pelo algoritmo de AutoML proposto.

Problema	α	γ
swiss42	0,671	0,306
eil51	0,693	0,352
berlin52	0,684	0,003
st70	0,666	0,311
eil76	0,673	0,348
pr76	0,990	0,900
rat99	0,688	0,326
kroa100	0,674	0,340
eil101	0,692	0,357
bier127	0,681	0,225
ch130	0,990	0,150
ftv33	0,653	0,037
p43	0,990	0,010
ftv44	0,680	0,341
ftv47	0,681	0,309
ry48p	0,684	0,314
ft53	0,707	0,400
ftv64	0,679	0,353
ft70	0,799	0,222

parâmetros sendo necessária a realização de uma avaliação para realizar o ajuste de parâmetros necessário para cada problema avaliado.

A Fig. 3 exibe a rota final obtida para a instância st70 aplicando os parâmetros definidos de maneira automática na segunda etapa do experimento, neste caso, os parâmetros utilizados foram os pontos estacionários do modelo ajustado para à superfície de resposta gerada. Vale ressaltar quê este gráfico foi obtido automaticamente através do algoritmo proposto durante a última etapa do experimento.

4.4 Comparação com parâmetros de outros trabalhos

A Tabela 8 apresenta os resultados obtidos em uma nova rodada de experimentos. Para legitimar os resultados já apresentados e a importância do assunto que é estudado neste artigo, decidiu-se pela realização de uma nova rodada de experimentos. Para tal, foram selecionados alguns trabalhos que aplicaram o Aprendizado por Reforço no Problema do Caixeiro Viajante. Nesse sentido, foram selecionados os trabalhos de Gambardella and Dorigo (1995), Sun et al. (2001), Liu and Zeng (2009) e Santos (2009), e destes trabalhos foram extraídos os seguintes parâmetros:

- $\alpha = 0,1$ e $\gamma = 0,3$ (Gambardella and Dorigo, 1995).
- $\alpha = 0,1$ e $\gamma = 0,9$ (Liu and Zeng, 2009).
- $\alpha = 0,8$ e $\gamma = 0,9$ (Sun et al., 2001).
- $\alpha = 0,8$ e $\gamma = 1$ (Santos, 2009).

Para à realização desta novo ciclo, foram executadas 5 épocas sendo que cada época contou com 10.000 episódios. Cada um dos parâmetros selecionados foram testados nessas condições. Além disso, neste novo ciclo foram utilizados também os parâmetros definidos pelo sistema proposto, sendo que estes parâmetros são apresentados

Tabela 8: Resultados da aplicação dos parâmetros definidos pelo sistema de AutoML e dos parâmetros selecionados de outros trabalhos, sendo **AutoML:** resultados obtidos com os parâmetros definidos através do sistema de AutoML; **S09:** resultados com parâmetros de Santos (2009); **G95:** resultados com parâmetros de Gambardella and Dorigo (1995); **L09:** resultados com parâmetros de Liu and Zeng (2009); **S01:** resultados com parâmetros de Sun et al. (2001).

Problema	Ótimo	AutoML	S09	G95	L09	S01
swiss42	1273	**1335**	1689	1364	1564	1435
eil51	426	**475**	614	478	505	497
berlin52	7542	**7871**	9620	8422	8731	8665
st70	675	**704**	1032	712	765	780
eil76	538	**567**	791	570	587	595
pr76	108159	120665	150757	123123	119189	**117914**
rat99	1211	**1350**	2093	1373	1372	1419
kroa100	21282	24555	35347	**24278**	27552	26626
eil101	629	**707**	981	732	728	716
bier127	118282	**126922**	169424	129382	144413	146748
ch130	6110	**6590**	7659	6854	7702	7516
ftv33	1286	**1365**	1795	1472	1556	1578
p43	5620	**5637**	5647	5652	5651	5648
ftv44	1613	1852	2755	1876	**1850**	1867
ftv47	1776	**2087**	2993	2100	2266	2157
ry48p	14422	**15575**	19899	**15575**	16775	16480
ft53	6905	8182	9263	8287	7945	**7895**
ftv64	1839	**2100**	3433	2130	2333	2286
ft70	38673	41568	47847	**41382**	42533	42040

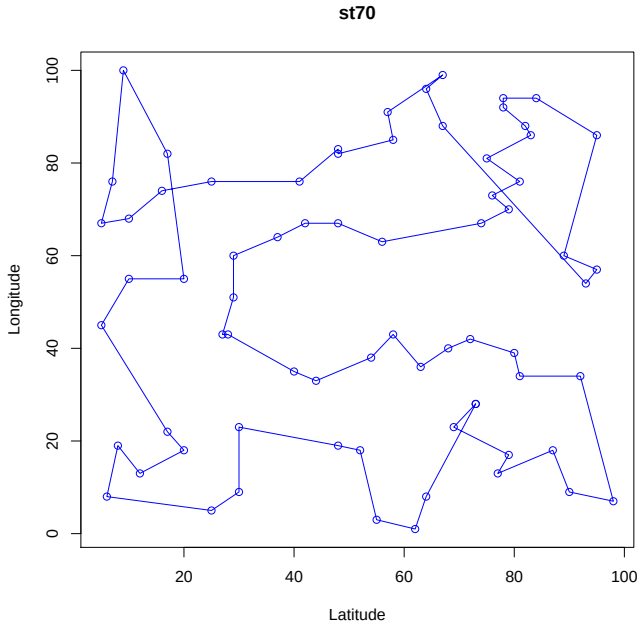

Figura 3: Rota final obtida para o problema st70 com o algoritmo de AutoML proposto utilizando os parâmetros $\alpha = 0,666$ e $\gamma = 0,311$. Distância final: 707.

na Tabela 7. Visando assim uma comparação imparcial e justa dentre todos os parâmetros.

Como forma de organização, alguns ajustes foram realizados nos títulos das colunas da Tabela 8. Nessa linha, foi utilizada a seguinte regra para definição dos títulos das colunas que apresentam os resultados obtidos com os parâmetros de outros trabalhos: primeira letra do autor seguido dos dois últimos dígitos correspondentes ao ano do trabalho. Assim, Santos (2009) será representado por 'S09', Gambardella and Dorigo (1995) por 'G95', Liu and Zeng (2009) por 'L09' e Sun et al. (2001) por 'S01'. No que se refere a coluna que apresenta os resultados obtidos com os parâmetros definidos com o sistema de AutoML proposto por este trabalho, esta foi nomeada com o título de 'AutoML'.

A partir dos resultados apresentados na Tabela 8, pode-se perceber que as instâncias swiss42, eil51, berlin52, st70, eil76, rat99, eil101, berlin127, ch130, ftv33, p43, ftv47, ftv64 obtiveram um melhor resultado quando foram aplicados os parâmetros definidos pelo sistema de AutoML. Já quando foram aplicados os parâmetros de Gambardella and Dorigo (1995), as instâncias kroa100 e ft70, apresentaram um melhor desempenho. Quando foram utilizados os parâmetros de Sun et al. (2001), obteve-se o melhor resultado em pr76 e ft53. Ao utilizar os parâmetros de Liu and Zeng (2009), a instância ftv44 apresentou o melhor desempenho. Por fim, tratando-se dos parâmetros extraídos de Santos (2009), nenhuma instância apresentou o melhor resultado. Não menos importante, vale ressaltar ainda quê a instância ry48p convergiu igualmente para o melhor resultado tanto quando foram aplicados os parâmetros definidos pelo sistema quanto quando foram aplicados os parâmetros selecionados do trabalho de Gambardella and Dorigo (1995).

5 Conclusão

O objetivo deste trabalho foi propor um sistema de Aprendizado por Reforço Automatizado (AutoRL-TSP-RSM) com Metodologia de Superfície de Resposta para o Problema do Caixeiro Viajante. O sistema proposto tem como intuito ajustar de forma automática os parâmetros de taxa de aprendizado e fator de desconto do algoritmo Q-

learning. Para isso, foram utilizados conceitos e realizados avanços em relação a trabalhos recentes (Ottoni et al., 2018, Ottoni, Ottoni, Oliveira and Nepomuceno, 2020).

A partir do sistema de AutoRL-TSP-RSM proposto foi possível ajustar parâmetros para 20 instâncias da TSPLIB de forma automática. Os resultados revelaram que, em geral, as menores de distâncias de rota foram alcançadas ao aplicar os valores definidos pelo sistema de AutoML, em comparação com parâmetros definidos na literatura.

Em trabalhos futuros, espera-se experimentar o sistema desenvolvido com outros problemas da TSPLIB e outros problemas de otimização combinatória. Além disso, espera-se ainda evoluir o sistema para indicação de algoritmos e para o ajuste do parâmetro Política ϵ-greedy.

Referências Bibliográficas

Alipour, M., Razavi, S., Feizi Derakhshi, M. and Balafar, M. (2018). A hybrid algorithm using a genetic algorithm and multiagent reinforcement learning heuristic to solve the traveling salesman problem, *Neural Computing and Applications* **30**(9): 2935–2951. http://dx.doi.org/10.1007/s00521-017-2880-4.

Alzubaidi, L., Zhang, J., Humaidi, A., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M., Al-Amidie, M. and Farhan, L. (2021). Review of deep learning: concepts, cnn architectures, challenges, applications, future directions, *Journal of Big Data* **8**(1). http://dx.doi.org/10.1186/s40537-021-00444-8.

Bianchi, R. A. C., Ribeiro, C. H. C. and Costa, A. H. R. (2009). On the relation between ant colony optimization and heuristically accelerated reinforcement learning, *1st international workshop on hybrid control of autonomous system*, Citeseer, pp. 49–55. Disponível em http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.605.9537&rep=rep1&type=pdf.

Bianchi, R. A. d. C. (2004). *Uso de heurísticas para a aceleração do aprendizado por reforço.*, PhD thesis, Universidade de São Paulo. https://doi.org/10.11606/T.3.2004.tde-28062005-191041.

Bodin, L. (1983). Routing and scheduling of vehicles and crews, the state of the art, *Comput. Oper. Res.* **10**(2): 63–211. https://doi.org/10.1016/0305-0548(83)90030-8.

Boeing, D. H. A. et al. (2019). Ensinando um robô a julgar: pragmática, discricionariedade e vieses no uso de aprendizado de máquina no judiciário. Disponível em https://repositorio.ufsc.br/bitstream/handle/123456789/203514/TCC-Ensinandoumrobôajulgar1-3-merged.pdf?sequence=1&isAllowed=y.

Brazdil, P., Carrier, C. G., Soares, C. and Vilalta, R. (2008). *Metalearning: Applications to data mining*, Springer Science & Business Media. https://doi.org/10.1007/978-3-540-73263-1.

Cai, H., Lin, J., Lin, Y., Liu, Z., Wang, K., Wang, T., Zhu, L. and Han, S. (2020). Automl for architecting efficient and specialized neural networks, *IEEE Micro* **40**(1): 75–82. https://doi.org/10.1109/MM.2019.2953153.

Celiberto Jr, L. A. (2007). *Aprendizado por reforço acelerado por heurísticas no domínio do futebol de robôs simulado*, Master's thesis, Centro Universitário da FEI. Disponível em https://repositorio.fei.edu.br/bitstream/FEI/437/1/fulltext.pdf.

Chen, S., Wu, J. and Chen, X. (2019). Deep reinforcement learning with model-based acceleration for hyperparameter optimization, *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 170–177. https://doi.org/10.1109/ICTAI.2019.00032.

Espindola, G. M. d. (2009). Uso de algoritmos genéticos no ajuste de parâmetros da segmentação de imagens, *XIV Simpósio Brasileiro de Sensoriamento Remoto*, pp. 6861–6868. Disponível em http://marte.sid.inpe.br/col/dpi.inpe.br/sbsr@80/2008/11.01.20.18/doc/6861-6868.pdf.

Faria, G. (2000). *Explorando o potencial de algoritmos de aprendizado com reforço em robôs móveis*, Master's thesis, Universidade de São Paulo. https://doi.org/10.11606/D.55.2020.tde-19022020-091603.

Feitosa, R. Q., Costa, G. A. O. P., Fredrich, C. M. B., Camargo, F. F. and de Almeida, C. M. (2009). Uma avaliação de métodos genéticos para ajuste de parâmetros de segmentação, *XIV Simpósio Brasileiro de Sensoriamento Remoto*, pp. 6875–6882. Disponível em http://marte.sid.inpe.br/col/dpi.inpe.br/sbsr@80/2008/11.18.13.07/doc/6875-6882.pdf?languagebutton=pt-BR.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M. and Hutter, F. (2015). Efficient and robust automated machine learning, *in* C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama and R. Garnett (eds), *Advances in Neural Information Processing Systems 28*, Curran Associates, Inc., pp. 2962–2970. Disponível em https://proceedings.neurips.cc/paper/2015/file/11d0e6287202fced83f79975ec59a3a6-Paper.pdf.

Gambardella, L. M. and Dorigo, M. (1995). Ant-q: A reinforcement learning approach to the traveling salesman problem, *Proceedings of the Twelfth International Conference on Machine Learning* pp. 252–260. https://doi.org/10.1016/B978-1-55860-377-6.50039-6.

Gershman, S. J. (2016). Empirical priors for reinforcement learning models, *Journal of Mathematical Psychology* **71**: 1–6. https://doi.org/10.1016/j.jmp.2016.01.006.

Goldbarg, M. C. and Luna, H. P. L. (2005). *Otimização combinatória e programação linear: modelos e algoritmos*, Elsevier.

Hutter, F., Hoos, H. and Leyton-Brown, K. (2014). An efficient approach for assessing hyperparameter importance, *Proceedings of International Conference on Machine Learning 2014 (ICML 2014)*, pp. 754–762. "Disponível em http://proceedings.mlr.press/v32/hutter14.pdf.

Hutter, F., Kotthoff, L. and Vanschoren, J. (eds) (2018). *Automated Machine Learning: Methods, Systems, Challenges*, Springer. https://doi.org/10.1007/978-3-030-05318-5.

Júnior, F. C. D. L., Neto, A. D. D. and De Melo, J. D. (2010). Hybrid metaheuristics using reinforcement learning applied to salesman traveling problem, *Traveling Salesman Problem, Theory and Applications*, IntechOpen. https://doi.org/10.5772/13343.

Lakshmi, E. S., Rao, M. N. and Sudhamani, M. (2020). Response surface methodology-artificial neural network based optimization and strain improvement of cellulase production by streptomyces sp., *Bioscience Journal* **36**(4): 1390–1402. Disponível em https://docs.bvsalud.org/biblioref/2021/02/1147303/48006-article-text-229344-1-10-20200527.pdf.

Leotti, V. B., Birck, A. R. and Riboldi, J. (2005). Comparação dos testes de aderência à normalidade kolmogorov-smirnov, anderson-darling, cramer–von mises e shapiro-wilk por simulação, *Anais do 11º Simpósio de Estatística Aplicada à Experimentação Agronômica* . Disponível em https://www.inf.ufsc.br/~vera.carmo/Testes_de_Hipoteses/Testes_aderencia.pdf.

Liu, F. and Zeng, G. (2009). Study of genetic algorithm with reinforcement learning to solve the tsp, *Expert Systems with Applications* **36**(3): 6995–7001. https://doi.org/10.1016/j.eswa.2008.08.026.

Makmal, A., Melnikov, A. A., Dunjko, V. and Briegel, H. J. (2016). Meta-learning within projective simulation, *IEEE Access* **4**: 2110–2122. https://doi.org/10.1109/access.2016.2556579.

Mantovani, R. G., Horváth, T., Cerri, R., Vanschoren, J. and de Carvalho, A. C. (2016). Hyper-parameter tuning of a decision tree induction algorithm, *5th Brazilian Conference on Intelligent Systems (BRACIS)*, IEEE, pp. 37–42. https://doi.org/10.1109/BRACIS.2016.018.

Mantovani, R. G., Rossi, A. L., Alcobaça, E., Vanschoren, J. and de Carvalho, A. C. (2019). A meta-learning recommender system for hyperparameter tuning: Predicting when tuning improves svm classifiers, *Information Sciences* **501**: 193–221. https://doi.org/10.1016/j.ins.2019.06.005.

Mantovani, R. G., Rossi, A. L., Vanschoren, J., Bischl, B. and Carvalho, A. C. (2015). To tune or not to tune: recommending when to adjust svm hyper-parameters via meta-learning, *2015 International Joint Conference on Neural Networks (IJCNN)*, IEEE, pp. 1–8. https://doi.org/10.1109/IJCNN.2015.7280644.

Martins, M. F. (2007). *Aprendizado por reforço acelerado por heurísticas aplicado ao domínio do futebol de robôs*, Master's thesis, Centro Universitário da FEI. Disponível em https://repositorio.fei.edu.br/bitstream/FEI/417/2/fulltext.pdf.

Mitchell, T. M. (1997). *Machine Learning*, McGraw-hill New York.

Monard, M. C. and Baranauskas, J. A. (2003). Conceitos sobre aprendizado de máquina, *Sistemas Inteligentes Fundamentos e Aplicações*, 1 edn, Manole Ltda, Barueri-SP, pp. 39–56. Disponível em https://dcm.ffclrp.usp.br/~augusto/publications/2003-sistemas-inteligentes-cap4.pdf.

Myers, R. H., Montgomery, D. C. and Anderson-Cook, C. M. (2016). *Response surface methodology: process and product optimization using designed experiments*, John Wiley & Sons.

Ottoni, A. L. C. (2016). *Análise de sensibilidade dos parâmetros do aprendizado por reforço na solução do problema do caixeiro viajante*, Master's thesis, Programa de Pós-Graduação em Engenharia Elétrica da associação ampla CEFET-MG e UFSJ. Disponível em https://drive.google.com/open?id=0B33h6pvItVsQYktld2d6alU5RHM.

Ottoni, A. L. C., Nepomuceno, E. G., Cordeiro, L. T., Lamperti, R. D. and Oliveira, M. (2015). Análise do desempenho do aprendizado por reforço na solução do problema do caixeiro viajante, *XII SBAI-Simpósio Brasileiro de Automação Inteligente* pp. 43–48. Disponível em http://swge.inf.br/SBAI2015/anais/017.pdf.

Ottoni, A. L. C., Nepomuceno, E. G. and de Oliveira, M. S. (2016). Aprendizado por reforço na solução do problema do caixeiro viajante assimétrico: Uma comparação entre os algoritmos q-learning e sarsa, *XII Simpósio de Mecânica Computacional*. Disponível em https://www.ufsj.edu.br/portal2-repositorio/File/gcom/OttoniSIMMEC2016.pdf.

Ottoni, A. L. C., Nepomuceno, E. G. and de Oliveira, M. S. (2017). Análise do desempenho do aprendizado por reforço na solução do problema da mochila multidimensional, *Revista Brasileira de Computação Aplicada* **9**(3): 56–70. https://doi.org/10.5335/rbca.v9i3.6601.

Ottoni, A. L. C., Nepomuceno, E. G., de Oliveira, M. S. and de Oliveira, D. C. R. (2020). Tuning of reinforcement learning parameters applied to sop using the scott–knott method, *Soft Computing* **24**(6): 4441–4453. https://doi.org/10.1007/s00500-019-04206-w.

Ottoni, A. L. C., Nepomuceno, E. G. and Oliveira, M. S. (2016). Análise de sensibilidade dos parâmetros do aprendizado por reforço na solução do problema do caixeiro viajante: modelagem via superfície de resposta, *XXI Congresso Brasileiro de Automática*, Vol. 21, pp. 513–518. Disponível em http://www.swge.inf.br/PDF/CBA2016-0170_047213.PDF.

Ottoni, A. L. C., Nepomuceno, E. G. and Oliveira, M. S. (2019). Estimação de parâmetros do aprendizado por reforço para o problema de planejamento de rotas com reabastecimento, *Simpósio Brasileiro de Automação Inteligente*. https://doi.org/10.17648/sbai-2019-111113.

Ottoni, A. L. C., Nepomuceno, E. G. and Oliveira, M. S. d. (2018). A response surface model approach to parameter estimation of reinforcement learning for the travelling salesman problem, *Journal of Control, Automation and Electrical Systems* **29**(3): 350–359. https://doi.org/10.1007/s40313-018-0374-y.

Ottoni, L. T. C., Ottoni, A. L. C., Oliveira, M. S. d. and Nepomuceno, E. G. (2020). Autorl-tsp: Sistema de aprendizado por reforço automatizado para o problema do caixeiro viajante, *XXIII Congresso Brasileiro de Automática*. https://doi.org/10.48011/asba.v2i1.1658.

Pedro, O. R. (2013). *Uma abordagem de busca tabu para o problema do caixeiro viajante com coleta de prêmios*, Master's thesis, Universidade Federal de Minas Gerais. Disponível em https://www.ppgee.ufmg.br/defesas/962M.PDF.

Pellegrini, J. and Wainer, J. (2007). Processos de decisão de markov: um tutorial, *Revista de Informática Teórica e Aplicada* **14**(2): 133–179. https://doi.org/10.22456/2175-2745.5694.

Razali, N. M., Wah, Y. B. et al. (2011). Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests, *Journal of Statistical Modeling and Analytics* **2**(1): 21–33. Disponível em https://www.nbi.dk/~petersen/Teaching/Stat2019/Power_Comparisons_of_Shapiro-Wilk_Kolmogorov-Smirn.pdf.

Reinelt, G. (1991). Tsplib—a traveling salesman problem library, *ORSA journal on computing* **3**(4): 376–384. https://doi.org/10.1287/ijoc.3.4.376.

Reinelt, G. (1995). Tsplib95, *University Heidelberg* . Disponível em http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp95.pdf.

Riboldi, J. and Nascimento, L. d. C. S. C. d. (1994). Metodologia de superfície de resposta: uma abordagem introdutória, *Cadernos de matemática e estatística. Série B, Trabalho de apoio didático. Porto Alegre. No. 25 (nov. 1994), 83 f.* . https://doi.org/10183/205091.

Rossi, A. L. D. (2009). *Ajuste de parâmetros de técnicas de classificação por algoritmos bioinspirados*, Master's thesis, Universidade de São Paulo. https://doi.org/10.11606/D.55.2009.tde-06052009-114528.

Rossi, R. G. (2015). *Classificação automática de textos por meio de aprendizado de máquina baseado em redes*, PhD thesis, Universidade de São Paulo. https://doi.org/10.11606/T.55.2016.tde-05042016-105648.

Russell, S. J. and Norvig, P. (2013). *Inteligência artificial*, Campus, 3st ed.

Santos, J. P. Q. d. (2009). *Uma implementação paralela híbrida para o problema do caixeiro viajante usando algoritmos genéticos, grasp e aprendizagem por reforço*, Master's thesis, Universidade Federal do Rio Grande do Norte. Disponível em https://repositorio.ufrn.br/bitstream/123456789/15221/1/JoaoPQS.pdf.

Santos, J. P. Q. d. (2014). *Estratégias de busca reativa utilizando aprendizagem por reforço e algoritmos de busca local*, PhD thesis, UFRN. Disponível em https://repositorio.ufrn.br/bitstream/123456789/19393/1/JoaoPauloQueirozDosSantos_TESE.pdf.

Santos, J. P. Q. d., de Melo, J. D., Neto, A. D. D. and Aloise, D. (2014). Reactive search strategies using reinforcement learning, local search algorithms and variable neighborhood search, *Expert Systems with Applications* **41**(10): 4939–4949. https://doi.org/10.1016/j.eswa.2014.01.040.

Serra, M. R. G. (2004). *Aplicações de aprendizagem por reforço em controle de tráfego veicular urbano*, Master's thesis, Universidade Federal de Santa Catarina. Disponível em https://repositorio.ufsc.br/bitstream/handle/123456789/87535/206482.pdf?sequence=1&isAllowed=y.

Silva, A. L. M. (2014). *Algoritmo baseado em evolução diferencial para solução de problemas de otimização combinatória*, Master's thesis, Universidade Federal de Minas Gerais. Disponível em https://www.ppgee.ufmg.br/defesas/1039M.PDF.

Stamoulis, D., Ding, R., Wang, D., Lymberopoulos, D., Priyantha, N. B., Liu, J. and Marculescu, D. (2020). Single-path mobile automl: Efficient convnet design and nas hyperparameter optimization, *IEEE Journal of Selected Topics in Signal Processing* pp. 609 – 622. https://doi.org/10.1109/JSTSP.2020.2971421.

Stange, R. L. (2011). *Adaptatividade em aprendizagem de máquina: conceitos e estudo de caso.*, Master's thesis, Universidade de São Paulo. https://doi.org/10.11606/D.3.2011.tde-02072012-175054.

Sun, R., Tatsumi, S. and Zhao, G. (2001). Multiagent reinforcement learning method with an improved ant colony system, *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat. No. 01CH37236)*, Vol. 3, IEEE, pp. 1612–1617. https://doi.org/10.1109/ICSMC.2001.973515.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences, *Machine learning* **3**(1): 9–44. https://doi.org/10.1007/BF00115009.

Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding, *Advances in neural information processing systems*, pp. 1038–1044. Disponível em https://proceedings.neurips.cc/paper/1995/file/8f1d43620bc6bb580df6e80b0dc05c48-Paper.pdf.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*, MIT press, 2nd ed.

Tsiakmaki, M., Kostopoulos, G., Kotsiantis, S. and Ragos, O. (2019). Implementing automl in educational data mining for prediction tasks, *Applied Sciences* **10**(1). https://doi.org/10.3390/app10010090.

Tuggener, L., Amirian, M., Rombach, K., Lörwald, S., Varlet, A., Westermann, C. and Stadelmann, T. (2019). Automated machine learning in practice: State of the art and recent results, *2019 6th Swiss Conference on Data Science (SDS)*, pp. 31–36. https://doi.org/10.1109/SDS.2019.00-11.

Vitor, A. (2015). *Uma proposta de algoritmo genético híbrido para o problema do caixeiro viajante*, PhD thesis, Universidade Federal do Paraná. https://doi.org/1884/41345.

Wang, H., Cui, Z., Sun, H., Rahnamayan, S. and Yang, X.-S. (2017). Randomly attracted firefly algorithm with neighborhood search and dynamic parameter adjustment mechanism, *Soft Computing* **21**(18): 5325–5339. https://doi.org/10.1007/s00500-016-2116-z.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*, PhD thesis, King's College. Disponível em http://www.cs.rhul.ac.uk/~chrisw/new_thesis.pdf.

Watkins, C. J. and Dayan, P. (1992). Q-learning, *Machine learning* **8**(3-4): 279–292. https://doi.org/10.1007/BF00992698.

Zhao, J.-h., Li, F. and Zhang, X.-x. (2012). Parameter adjustment based on improved genetic algorithm for cognitive radio networks, *The Journal of China Universities of Posts and Telecommunications* **19**(3): 22–26. https://doi.org/10.1016/S1005-8885(11)60260-4.